AD-A093 618 STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB F/G 12/1
WORST CASE ANALYSIS OF GREEDY HEURISTICS FOR INTEGER PROGRAMMIN--ETC(U)
OCT 80 G DOBSON N00014-75-C-0267
UNCLASSIFIED SOL-80-25 NL

1 OF 1
AD A
093618

END
DATE
FILMED
2-8
DTIC

LEVEL II

Systems
Optimization
Laboratory

AD A093618

DDC FILE COPY

DTIC
ELECTE
JAN 0 9 1981
S D
E

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Department of Operations Research
Stanford University
Stanford, CA 94305

81 1 08 069

LEVEL II

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

WORST CASE ANALYSIS OF GREEDY HEURISTICS FOR INTEGER PROGRAMMING WITH NON-NEGATIVE DATA

by

Greg Dobson

TECHNICAL REPORT SOL 80-25

October 1980

Research and reproduction of this report were supported by the Department of Energy Contract DE-AC03-76SF00326, PA# DE-AT03-76ER72018, Office of Naval Research Contract N00014-75-C-0267, National Science Foundation Grants MCS76-81259, MCS-7926009 and ENG77-06761.

Worst Case Analysis of
Greedy Heuristics for Integer Programming with Non-negative Data

Greg Dobson
Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, California 94305

Abstract

We give a worst case analysis for two greedy heuristics for the integer programming problem minimize cx, $Ax \geq b$, $0 \leq x \leq u$, x integer, where the entries in A, b, and c are all non-negative. The first heuristic is for the case where the entries in A and b are integral, the second only assumes the rows are scaled so that the smallest nonzero entry is at least 1. In both cases we compare the ratio of the value of the greedy solution to that of the integer optimal. The error bound grows logarithmically in the maximum column sum of A for both heuristics.

---

1. Introduction

Consider the integer programming problem

$$\begin{aligned} &\text{minimize } cx \\ &\quad Ax \geq b \\ &\quad 0 \leq x \leq u \quad x \text{ integer} \end{aligned} \tag{P}$$

with the additional restriction on the data that it be nonnegative (i.e. $a_{ij} \geq 0$, $b_i \geq 0$, $c_j \geq 0$ for all i, j). We establish tight bounds on the worst case behavior of two greedy heuristics. If the constraint data is either integral or if each row is scaled so that the smallest entry is at least 1, then the error bound tends to grow logarithmically in the maximum column sum in "A". The results here are a direct generalization of the work of Chvátal[1] on the set covering problem and of Lovász[4] and Johnson[3] on the unit cost set covering problem.

In § 2 we describe the basic algorithm and give a *proof for* the case of integral constraint data and no upper bounds. In § 3 the analysis is extended to the case where upper bounds are allowed. § 4 deals with the case of non-integral constraint data. Here it is necessary to introduce a modification of our basic algorithm to maintain a logarithmic error bound. Finally § 5 gives some evidence why the non-negativity of the data cannot be dropped and still have an approximation algorithm that runs in polynomial time. This section also shows why certain improvements in "greedy" heuristics cannot provide better than logarithmic error bounds.

2. The Basic Algorithm and Error Analysis

Throughout this section we assume that the constraint data is integral and that there are no upper bounds on the variables.

The greedy heuristic picks the column j^* that minimizes $c_j / \sum_{i=1}^{m} a_{ij}$, increments x_{j^*} by 1 and repeats the process. Column j^* minimizes the myopic unit cost of satitfying the constraints. Observe that when we have $a_{ij} > b_i$ then setting $x_j = 1$ would satisfy the i'th constraint yet this "large a_{ij}" value could make the ratio $c_j / \sum_{i=1}^{m} a_{ij}$ appear artificially small. In order to obtain any bound at all it will be necessary to adjust the matrix data during the algorithm to elliminate "large a_{ij}'s". In particular at the end of every iteration we adjust any $a_{ij} > b_i$ down to b_i, i.e. $a_{ij} \leftarrow \min(a_{ij}, b_i)$. For simplicity we assume this adjustment has been made to the original data so $a_{ij} \leq b_i$.

The greedy heuristic is

Greedy 1 (no upper bounds)

$x \leftarrow 0$
$z \leftarrow 0$
while $b \neq 0$ do
begin
$k \leftarrow \operatorname{argmin}_{1 \le j \le n}\left(c_j / \sum_{i=1}^{m} a_{ij}\right)$
$x_k \leftarrow x_k + 1$
$b_i \leftarrow b_i - a_{ik}$ for all i
$a_{ij} \leftarrow \min(a_{ij}, b_i)$ for all i, j
$z \leftarrow z + c_k$
end

Denote by $H(d)$ the first d terms of the harmonic series: $H(d) = \sum_{i=1}^{d} 1/i$. We can make the following performance guarantee on the value of the heruristic solution:

Theorem 2.1. Given problem (P) with integral constraint data and no upper bounds on the variables, if x^* is the optimal integral solution and $\bar{x}$ is the solution given by the greedy heuristic then

$$\frac{c\bar{x}}{cx^*} \le H\left(\max_{1 \le j \le n} \sum_{i=1}^{m} a_{ij}\right)$$

and this bound is tight.

To see that the bound is tight consider the program :

$$\begin{array}{llllllll}
\text{minimize} & \frac{1}{d}x_1 + & \frac{1}{d-1}x_2 + & \cdots + & \frac{1}{2}x_{d-1} + & \frac{1}{1}x_d + & (1+\epsilon)x_{d+1} & \\
& x_1 & & & & + & x_{d+1} & \ge 1 \\
& & x_2 & & & + & x_{d+1} & \ge 1 \\
& & & & & \vdots & & \vdots \\
& & & & x_{d-1} & + & x_{d+1} & \ge 1 \\
& & & & & x_d + & x_{d+1} & \ge 1
\end{array}$$

where $x_j = 0, 1$. The heuristic picks the solution $\bar{x} = (1, \ldots, 1, 0)$ whereas the optimal solution is $x^* = (0, \ldots, 0, 1)$ for every $\epsilon > 0$. In this case as $\epsilon \to 0$

$$\frac{c\bar{x}}{cx^*} = \frac{H(d)}{1+\epsilon} \to H(d).$$

Because the data (A, b) is changing throughout the algorithm, we introduce the notation $A^r = (a^r_{ij})$ and $b^r = (b^r_i)$ to refer to the data at the start of the

r'th iteration. Let $w_j^r = \sum_{i=1}^m a_{ij}^r$, that is the j'th column sum at iteration r. Assume that the algorithm terminates after t iterations, that is $b^{t+1} = 0$. At each iteration r, it picks column k_r, x_{k_r} is increased by 1, so "b_i" is decreased by $a_{ik_r}^r$. Thus $b_i^{r+1} = b_i^r - a_{ik_r}^r$ or $a_{ik_r}^r = b_i^r - b_i^{r+1}$.

Note that if column k_r is picked, it is picked at least $p \equiv \min_{1 \le i \le m} \lfloor b_i^r / a_{ik_r}^r \rfloor$ times consecutively. To see this observe that the p'th time will be the first time that one of the a_{ik_r}'s will be reduced, thus the first time $\sum_{i=1}^m a_{ik_r}$ will be reduced. Even though the other column sums are (possibly) being reduced by the b_i's, that only makes those columns less attractive. Therefore any good implementation would always increment x_{k_r} by p. Suppose we implement the algorithm this way. Once x_{k_r} is choosen $b_{\hat{i}}^{r+1} = a_{\hat{i}k_r}^{r+1}$ where $\hat{i} \equiv \operatorname{argmin}_{1 \le i \le m} \lfloor b_i^r / a_{ik_r}^r \rfloor$, thus if x_{k_r} is choosen again then row $\hat{i}$ is covered. Only m variables could be choosen twice and we have that the number of iterations is bounded by $n + m$; thus the heuristic is polynomial. For ease of notation we will use the former description of the heuristic.

To prove Theorem 2.1 we will need some machinery provided by the lemmas below. In particular we introduce two sets of "price" functions which will be useful in comparing the value of the heuristic solution and all other feasible solutions.

Thus introduce a set of step functions $p_i(s)$ that will represent prices paid per unit by the greedy heuristic to satisfy the constraints. We wish to view satisfying the i'th constraint ($\sum_{i=1}^m a_{ij}x_j \ge b_i$) as covering up the interval $[0, b_i)$. In particular we will cover it up from right to left so that the interval remaining to be covered at iteration r is $[0, b_i^r)$. Intuitively, for each point s in the interval $[0, b_i)$ the unit price paid by the greedy to cover this point is

$$p_i(s) = \frac{c_{k_r}}{w_{k_r}^r} \quad \text{if } s \in [b_i^{r+1}, b_i^r) \quad \text{for } r = 1, \ldots, t$$

That it is indeed the exact price will now be shown.

Lemma 2.2.

$$c\bar{x} = \sum_{i=1}^m \int_{[0,b_i)} p_i(s)\, ds.$$

Proof.

$$\sum_{j=1}^{n} c_j x_j = \sum_{r=1}^{t} c_{k_r} = \sum_{r=1}^{t} \frac{c_{k_r}}{w_{k_r}^r} w_{k_r}^r$$
$$= \sum_{r=1}^{t} \frac{c_{k_r}}{w_{k_r}^r} \sum_{i=1}^{m} a_{k_r}^r = \sum_{i=1}^{m} \sum_{r=1}^{t} \frac{c_{k_r}}{w_{k_r}^r} (b_i^r - b_i^{r+1})$$
$$= \sum_{i=1}^{m} \sum_{r=1}^{t} \int_{[b_i^{r+1}, b_i^r)} p_i(s)\, ds = \sum_{i=1}^{m} \int_{[0,b_i)} p_i(s)\, ds. \qquad \blacksquare$$

We now define a unit price function $p_{ij}(s)$ for each element of the matrix analogous to the way we defined $p_i(s)$ for each b_i. Define

$$p_{ij}(s) = \begin{cases} \dfrac{c_j}{w_j^r}, & \text{if } s \in [a_{ij}^{r+1}, a_{ij}^r) \quad \text{for } r = 1, \ldots, t \\ p_i(s), & \text{if } s \in [a_{ij}^1, b_i) \end{cases}$$

We have that $p_{ij}(s)$ is non-increasing for s in $[0, b_i)$ since w_j^r is non-increasing in r. Intuitively $p_{ij}(s)$ is the price that would be paid to cover the point s in $[0, b_i)$ using column j. Because the heuristic is myopic we have

Lemma 2.3. If $s \in [0, b_i)$ then $p_i(s) \leq p_{ij}(s)$ for all j.

Proof. Fix j. Let r be the iteration number such that $s \in [a_{ij}^{r+1}, a_{ij}^r)$. Because "$a_{ij}$" was reduced at iteration r, we must have $a_{ij}^{r+1} = b_i^{r+1}$, and clearly $a_{ij}^r \leq b_i^r$, hence $s \in [b_i^{r+1}, b_i^r)$.

$$p_i(s) = \frac{c_{k_r}}{w_{k_r}^r} \leq \frac{c_j}{w_j^r} = p_{ij}(s)$$

where the inequality follows from the choice rule. $\blacksquare$

Theorem 2.4. Let $f\colon [0, b) \to [0, \infty)$ be non-increasing, $a \in (0, b]$, $S \subseteq [0, b)$, $\mu(S) \geq a$, then

$$\frac{1}{\mu(S)} \int_S f \leq \frac{1}{a} \int_{[0,a)} f.$$

Proof. Let $A = [0, a)$, then

$$\frac{1}{\mu(S)} \int_S f = \frac{\mu(S \cap A)}{\mu(S)} \left(\frac{1}{\mu(S \cap A)} \int_{S \cap A} f \right) + \frac{\mu(S - A)}{\mu(S)} \left(\frac{1}{\mu(S - A)} \int_{S-A} f \right).$$

This is a convex combination of two averages. Since f is non-increasing the first average is at least as large as the second. Thus a convex combination of these averages weighted more heavily on the first term can only be larger.

$$\leq \frac{\mu(S\bigcap A)}{\mu(A)}\left(\frac{1}{\mu(S\bigcap A)}\int_{S\bigcap A} f\right)+\frac{\mu(A-S)}{\mu(A)}\left(\frac{1}{\mu(S-A)}\int_{S-A} f\right).$$

Since f is non-increasing, f on $A-S$ is at least as large as f on $S-A$, thus $\frac{1}{\mu(S-A)}\int_{S-A} f \leq \frac{1}{\mu(A-S)}\int_{A-S} f$. After making this replacement in the second term we have

$$\leq \frac{\mu(S\bigcap A)}{\mu(A)}\left(\frac{1}{\mu(A\bigcap S)}\int_{A\bigcap S} f\right)+\frac{\mu(A-S)}{\mu(A)}\left(\frac{1}{\mu(A-S)}\int_{A-S} f\right)$$
$$=\frac{1}{\mu(A)}\int_A f.$$ ∎

We can now easily prove

Lemma 2.5.

$$\frac{a_{ij}}{b_i}\int_{[0,b_i)} p_i(s)\,ds \leq \int_{[0,a_{ij})} p_{ij}(s)\,ds.$$

Proof. First, since $p_i(s) \leq p_{ij}(s)$ for $s \in [0,b_i)$, by lemma 2.3, we have

$$\frac{1}{b_i}\int_{[0,b_i)} p_i(s)\,ds \leq \frac{1}{b_i}\int_{[0,b_i)} p_{ij}(s)\,ds.$$

Second, p_{ij} is non-increasing in s since w_j^r is non-increasing in r. Apply Theorem 2.4 with $S=[0,b_i)$ and $f=p_{ij}$ to obtain

$$\frac{1}{b_i}\int_{[0,b_i)} p_{ij}(s)\,ds \leq \frac{1}{a_{ij}}\int_{[0,a_{ij})} p_{ij}(s)\,ds.$$

Combining the last two inequalities the result is immediate. ∎

For the set covering problem where $a_{ij}=0$ or 1 and $b_i=1$ for all i,j a row i is either not covered, ($b_i^r=1$), or completely covered, ($b_i^r=0$), thus the price function $p_{ij}(s)$ only takes on one value, $\lambda_i = c_{k_r}/w_{k_r}^r$ where row i is covered at iteration r. In Chvátal's analysis of the greedy algorithm for the set covering problem he proved the inequality

$$\sum_{i=1}^m \lambda_i a_{ij} \leq c_j H\left(\sum_{i=1}^m a_{ij}\right) \quad \text{for } j=1,\ldots,n.$$

These are relaxed dual constraints of the associated linear programming problem. We are now in a position to prove the analogous inequality for this problem. For each j define $v_j \equiv \min\{r \mid w_j^{r+1}=0\}$.

Lemma 2.6.

$$\sum_{i=1}^{m} \frac{a_{ij}}{b_i} \int_{[0,b_i)} p_i(s)\, ds \le c_j h_j.$$

where

$$h_j = \sum_{r=1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r}.$$

Proof.

$$\begin{aligned}
\sum_{i=1}^{m} \frac{a_{ij}}{b_i} \int_{[0,b_i)} p_i(s)\, ds &\le \sum_{i=1}^{m} \int_{[0,a_{ij})} p_{ij}(s)\, ds \\
&= \sum_{i=1}^{m} \sum_{r=1}^{v_j} \frac{c_j}{w_j^r}(a_{ij}^r - a_{ij}^{r+1}) \\
&= c_j \sum_{r=1}^{v_j} \frac{1}{w_j^r}(w_j^r - w_j^{r+1}) \\
&= c_j h_j.
\end{aligned}$$

∎

With Lemmas 2.2 and 2.6 we now complete the proof of Theorem 2.1. By Lemma 2.2

$$c\bar{x} = \sum_{i=1}^{m} \int_{[0,b_i)} p_i(s)\, ds$$

Now let x be any feasible solution, so $\sum_{j=1}^{n} a_{ij}x_j \ge b_i$ or $\sum_{j=1}^{n} \frac{a_{ij}x_j}{b_i} \ge 1$, thus

$$\begin{aligned}
&\le \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \frac{a_{ij}x_j}{b_i} \right) \int_{[0,b_i)} p_i(s)\, ds \\
&= \sum_{j=1}^{n} \left(\sum_{i=1}^{m} \frac{a_{ij}}{b_i} \int_{[0,b_i)} p_i(s)\, ds \right) x_j
\end{aligned}$$

By lemma 2.6 we have

$$\begin{aligned}
&\le \sum_{j=1}^{n} (c_j h_j) x_j \\
&\le \left(\max_{1 \le j \le n} h_j \right) \left(\sum_{j=1}^{n} c_j x_j \right).
\end{aligned}$$

Thus,

$$\frac{c\bar{x}}{cx} \leq \max_{1\leq j\leq n} h_j$$

for all feasible x. We now use the assumption that the "a_{ij}'s" and "b_i's" are integral.

$$\begin{aligned} h_j &= \sum_{r=1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r} = \sum_{r=1}^{v_j} \sum_{w_j^{r+1}<l\leq w_j^r} \frac{1}{w_j^r} \\ &\leq \sum_{r=1}^{v_j} \sum_{w_j^{r+1}<l\leq w_j^r} \frac{1}{l} = \sum_{1\leq l\leq w_j^1} \frac{1}{l} \\ &= H(w_j^1) = H\left(\sum_{i=1}^{m} a_{ij}\right) \end{aligned}$$

Thus

$$\max_{1\leq j\leq n} h_j \leq \max_{1\leq j\leq n} H\left(\sum_{i=1}^{m} a_{ij}\right) = H\left(\max_{1\leq j\leq n} \sum_{i=1}^{m} a_{ij}\right)$$

Therefore, since the optimal solution, x^*, is feasible the final bound is

$$\frac{c\bar{x}}{cx} \leq \frac{c\bar{x}}{cx^*} \leq H\left(\max_{1\leq j\leq n} \sum_{i=1}^{m} a_{ij}\right). \quad \blacksquare$$

3. The Analysis with Upper Bounds

We now extend the previous result to the integer programming problem with upper bounds.

$$\begin{aligned} &\text{minimize } cx \\ &\qquad Ax \geq b \\ &\qquad 0 \leq x \leq u \quad x \text{ integer} \end{aligned} \tag{P}$$

Again we assume $a_{ij}, b_i, c_j \geq 0$ for all i, j. If $u = (1, \ldots, 1)$ we have the important special case of 0-1 variables. The primary difference in the algorithm is in finding that x_j has reached its upper bound u_j, the column A_j is set to 0 to prevent the column from being picked again. Because "a_{ij}" does not decrease with "b_i", i.e. $a_{ij}^r = \min(a_{ij}, b_i^r)$ does not always hold, lemmas 2.3 and 2.5 break down.

Greedy 1 (with upper bounds)
    $x \leftarrow 0$
    $z \leftarrow 0$
    while $b \neq 0$ do
        begin
            $k \quad \leftarrow \text{argmin}_{1 \le j \le n}(c_j / \sum_{i=1}^{m} a_{ij})$
            $x_k \quad \leftarrow x_k + 1$
            $b_i \quad \leftarrow b_i - a_{ik}$ for all i
            $a_{ij} \quad \leftarrow \min(a_{ij}, b_i)$ for all i, j
            $z \quad \leftarrow z + c_k$
            if $x_k = u_k$ then $A_k \leftarrow 0$
        end

The extension of Theorem 2.1 is

Theorem 3.1. Given problem (P) with integral constraint data and with optimal solution x^*, if $\bar{x}$ is the solution given by the greedy algorithm, then

$$\frac{c\bar{x}}{cx^*} \le H\left(\max_{1 \le j \le n} \sum_{i=1}^{m} a_{ij}\right).$$

Proof. For the moment we restrict our attention to constraint i, and let x be a vector that satisfies that constraint, $(\sum_{j=1}^{n} a_{ij}x_j \ge b_i)$. The difficulty arises for those variables $\bar{x}_j$ which reach their upper bounds u_j before the algorithm stops. Define $t_i \equiv \min\{r \mid b_i^{r+1} = 0\}$. Define $U_i \equiv \{j \mid \bar{x}_j = u_j \text{ but } k_{t_i} \neq j\}$, i.e. those variables that reached their upper bounds before the iteration that covered row i. For $j \in U_i$, define $W_{ij} \equiv \{\text{iterations } r \mid k_r = j\}$. W_{ij} is the set of iterations where column j is picked, and $\bar{x}_j$ is incremented.

Say $W_{ij} = \{s_1, \ldots, s_{u_j}\}$. Using the feasible x we define $V_{ij} \equiv \{s_1, \ldots, s_{x_j}\}$, i.e. the first "$x_j$" values from W_{ij}. We now divide the interval $[0, b_i)$ into two parts. On each part we compute a bound for the price paid by the greedy heuristic to cover that part. Define

$$R_i = \bigcup_{\substack{r \in V_{ij} \\ j \in U_i}} [b_i^{r+1}, b_i^r) \qquad S_i = [0, b_i) - R_i.$$

Claim:

$$\int_{R_i} p_i(s)\, ds \le \sum_{j \in U_i} \left(\int_{[0, a_{ij})} p_{ij}(s)\, ds \right) x_j \tag{3.1}$$

$$\frac{\min(a_{ij}, \mu(S_i))}{\mu(S_i)} \int_{S_i} p_i(s)\, ds \le \int_{[0, a_{ij})} p_{ij}(s)\, ds \qquad j \notin U_i \tag{3.2}$$

Proof of (3.1). First observe that for $j \in U_i$, "a_{ij}" only takes on the values 0 and a_{ij}^1. To see this assume $a_{ij}^1 \neq a_{ij}^r = b_i^r \neq 0$, that is a_{ij} took on another nonzero value. At this iteration $\bar{x}_j < u_j$ (otherwise $a_{ij}^r = 0$). Hence $j \in U_i$ implies column j is picked again and covers row i, $k_{t_i} = j$, thus $j \notin U_i$ contradiction.

$$\int_{R_i} p_i(s)\,ds = \sum_{j\in U_i}\sum_{r\in V_{ij}} \int_{[b_i^{r+1}, b_i^r)} p_i(s)\,ds$$

Now apply that fact that a_{ij} only takes on the values 0 and a_{ij}^1,

$$\begin{aligned} &= \sum_{j\in U_i}\sum_{r\in V_{ij}} \frac{c_j}{w_j^r} a_{ij}^1 \\ &\leq \sum_{j\in U_i}\left(\max_{r\in V_{ij}} \frac{c_j}{w_j^r}\right) a_{ij}^1 |V_{ij}| \\ &\leq \sum_{j\in U_i}\left(\max_{r\in W_{ij}} \frac{c_j}{w_j^r}\right) a_{ij}^1 x_j \end{aligned}$$

Notice that $\max_{r\in W_{ij}} c_j/w_j^r$ is the largest value of c_j/w_j^r before column j is set to zero and $w_j^{r+1} = 0$, and thus is the one that defines p_{ij} on $[0, a_{ij}^1)$.

$$= \sum_{j\in U_i}\left(\int_{[0,a_{ij})} p_{ij}(s)\,ds\right) x_j. \qquad \blacksquare$$

Proof of (3.2). Note $S_i \neq \emptyset$ since $[0, b_i^{t_i}) \subseteq S_i$. Define $\tilde{a}_{ij} \equiv \min(a_{ij}, \mu(S_i))$. Since $j \notin U_i$ lemma 2.3 holds, so

$$\frac{1}{\mu(S_i)}\int_{S_i} p_i(s)\,ds \leq \frac{1}{\mu(S_i)}\int_{S_i} p_{ij}(s)\,ds$$

by Theorem 2.4

$$\begin{aligned} &\leq \frac{1}{\tilde{a}_{ij}}\int_{[0,\tilde{a}_{ij})} p_{ij}(s)\,ds \\ &\leq \frac{1}{\tilde{a}_{ij}}\int_{[0,a_{ij})} p_{ij}(s)\,ds. \qquad \blacksquare \end{aligned}$$

We complete the proof of Theorem 3.1. $\sum_{j\notin U_i} a_{ij}x_j \geq b_i - \sum_{j\in U_i} a_{ij}x_j = b_i - \mu(R_i) = \mu(S_i)$. Recall that $\tilde{a}_{ij} = \min(a_{ij}, \mu(S_i))$, so $\sum_{j\notin U_i} \tilde{a}_{ij}x_j \geq \mu(S_i)$.

By Lemma 2.2 we have

$$\begin{aligned}
c\bar{x} &= \sum_{i=1}^{m} \int_{[0,b_i)} p_i(s)\,ds \\
&= \sum_{i=1}^{m} \int_{S_i} p_i(s)\,ds + \sum_{i=1}^{m} \int_{R_i} p_i(s)\,ds
\end{aligned}$$

If x is feasible $\sum_{j\notin U_i} \tilde{a}_{ij}x_j \geq \mu(S_i)$. thus

$$\leq \sum_{i=1}^{m}\Bigg(\sum_{j\notin U_i} \frac{\tilde{a}_{ij}x_j}{\mu(S_i)}\Bigg)\int_{S_i} p_i(s)\,ds + \sum_{i=1}^{m}\int_{R_i} p_i(s)\,ds$$

Applying (3.1) and (3.2)

$$\begin{aligned}
&\leq \sum_{i=1}^{m}\sum_{j\notin U_i}\Bigg(\int_{[0,a_{ij})} p_{ij}(s)\,ds\Bigg)x_j + \sum_{i=1}^{m}\sum_{j\in U_i}\Bigg(\int_{[0,a_{ij})} p_{ij}(s)\,ds\Bigg)x_j \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\Bigg(\int_{[0,a_{ij})} p_{ij}(s)\,ds\Bigg)x_j \\
&= \sum_{j=1}^{n}\Bigg(\sum_{i=1}^{m}\sum_{r=1}^{v_j} \frac{c_j}{w_j^r}(a_{ij}^r - a_{ij}^{r+1})\Bigg)x_j \\
&= \sum_{j=1}^{n}\Bigg(\sum_{r=1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r}\Bigg)c_jx_j
\end{aligned}$$

The rest follows as in the proof of theorem 2.1.

$$\leq H\Bigg(\max_{1\leq j\leq n}\sum_{i=1}^{m} a_{ij}\Bigg)\sum_{j=1}^{n} c_jx_j.$$

Finally if x^* is the optimal solution,

$$\frac{c\bar{x}}{cx} \leq \frac{c\bar{x}}{cx^*} \leq H\Bigg(\max_{1\leq j\leq n}\sum_{i=1}^{m} a_{ij}\Bigg). \qquad \blacksquare$$

4. Case of Non-integral Constraint Data

We now wish to drop the restriction on the integrality of the constraint data, A and b. Because we are now free to scale the rows so as to make a column sum as small as we like we need a standard form. We thus assume that each row $(a_{i1},\ldots,a_{in},b_i)$ has been scaled so that the smallest non-zero entry is at least

1. Unfortunately the error of the greedy heuristic can be as bad as linear in the number of rows as the following example will demonstrate.

$$\begin{array}{llllllllll}
\min \frac{1}{4}x_1 + \frac{1}{3}x_2 + \frac{1}{2}x_3 + \frac{1}{1}x_4 + x_5 + x_6 + x_7 + x_8 + & & & & \beta x_9 & \\
x_1 & + x_5 & & & + (1+\epsilon)x_9 & \geq (1+\epsilon) \\
\quad x_2 & & + x_6 & & + (1+\epsilon^2)x_9 & \geq (1+\epsilon^2) \\
\quad\quad x_3 & & & + x_7 & + (1+\epsilon^3)x_9 & \geq (1+\epsilon^3) \\
\quad\quad\quad x_4 & & & & + x_8 + (1+\epsilon^4)x_9 & \geq (1+\epsilon^4)
\end{array}$$

where $x_i = 0, 1$ for $i = 1, \ldots, 9$ and $\beta = 1 + \epsilon + \epsilon^2 + \epsilon^3 + \epsilon^4 + \epsilon^5$.

The heuristic picks $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$ in that order at which point the problem is reduced to

$$\begin{array}{lllll}
\text{minimize } x_5 + & x_6 + & x_7 + & x_8 + \beta x_9 & \\
\epsilon x_5 & & & + \epsilon x_9 & \geq \epsilon \\
& \epsilon^2 x_6 & & + \epsilon^2 x_9 & \geq \epsilon^2 \\
& & \epsilon^3 x_7 & + \epsilon^3 x_9 & \geq \epsilon^3 \\
& & & \epsilon^4 x_8 + \epsilon^4 x_9 & \geq \epsilon^4
\end{array}$$

Now the heuristic picks $x_5 = 1$, $x_6 = 1$, $x_7 = 1$, $x_8 = 1$, thus

$$\bar{x} = (1,1,1,1,1,1,1,1,0)$$

whereas you may verify that

$$x^* = (0,0,0,0,0,0,0,0,1)$$

so the error is

$$\frac{c\bar{x}}{cx^*} \approx H(4) + 4.$$

The extension to an arbitrary number of rows is obvious. In general the error can be shown to be

$$\frac{c\bar{x}}{cx^*} \leq \max_{1 \leq j \leq n} \left\{ \log\left(\sum_{i=1}^{m} a_{ij} \right) + 1 + d_j \right\}$$

where d_j is the number of non-zero entries in column j. But by a slight modification of the algorithm we can replace this d_j by $H(d_j)$ and regain a logarithmic error bound. The basic idea of how this is done can be seen in the last example. The main component of the error in the heuristic solution did not enter until an

element in a row (and thus all the elements in that row) was below 1. Once this occurs in all the remaining rows we can scale the reduced problem so it looks like a set covering problem. At this point Chvátal's original analysis (or Theorem 2.1) tells us the error is at most $H(\max_{1\leq j\leq n} d_j)$. Let δ be a fixed, arbitrarily small number.
The new algorithm is,

```
Greedy 2:
    x  ← 0
    z  ← 0
    while b ≠ 0 do
        begin
            for each j, if Σ_{i=1}^m a_ij < 1 then
              for any row i such that a_ij ≠ 0 has not been scaled so far,
              scale row i to δ.
            k    ← argmin_{1≤j≤n}(c_j / Σ_{i=1}^m a_ij)
            x_k  ← x_k + 1
            b_i  ← b_i − a_ik for all i
            a_ij ← min(a_ij, b_i) for all i, j
            z    ← z + c_k
            if x_k = u_k then A_k ← 0
        end
```

The main result is

Theorem 4.1. Consider the program

$$\begin{aligned} \text{minimize } & cx \\ & Ax \geq b \\ & 0 \leq x \leq u \quad x \text{ integer} \end{aligned} \tag{P}$$

where a_{ij}, b_i, $c_j \geq 0$, each row has been scaled so that the minimum non-zero entry is at least 1, and d_j is the number of non-zero entries in the j'th column of A. If x^* is the optimal solution and Greedy 2 is applied to (P) giving a solution $\bar{x}$ then

$$\frac{c\bar{x}}{cx^*} \leq \max_{1\leq j\leq n}\left\{\log\left(\sum_{i=1}^{m} a_{ij}\right) + 1 + H(d_j)\right\}.$$

Proof. Fix j. Recall from the proof of Theorem 3.1 that the only step that used the hypothesis of integrality of the constraint data was in showing that

$$\sum_{r=1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq H\left(\sum_{i=1}^{m} a_{ij}\right).$$

Thus to prove Theorem 4.1 it suffices to show that

$$\sum_{r=1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq \log\left(\sum_{i=1}^{m} a_{ij}\right) + 1 + H(d_j).$$

Let q be the iteration such that $w_j^q \geq 1$ but $w_j^{q+1} < 1$. We claim

$$\sum_{r=1}^{q} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq \log\left(\sum_{i=1}^{m} a_{ij}\right) + 1 \tag{4.1}$$

$$\sum_{r=q+1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq H(d_j). \tag{4.2}$$

The scaling, however, does cause one problem with the previous analysis. At some iteration r, a_{ij} may be scaled down to δ. Since p_{ij} and p_i reflect the prices paid by the greedy heuristic and the heuristic does not actually cover up the interval $[\delta, a_{ij}^r)$ we should not include this cost. To avoid this difficulty, we define $\bar{a}_{ij}^r$ to be the value of a_{ij} at iteration r before any scaling and a_{ij}^r the value after scaling, and similarly for $\bar{w}_j^r$ and w_j^r. If we no longer integrate over the intervals $[a_{ij}^r, \bar{a}_{ij}^r)$, the analysis goes through and we have

$$\frac{c\bar{x}}{cx^*} \leq \max_{1 \leq j \leq n} \sum_{r=1}^{v_j} \frac{w_j^r - \bar{w}_j^{r+1}}{w_j^r}.$$

Since $w_j^{r+1} \leq \bar{w}_j^{r+1}$ we may replace $\bar{w}_j^{r+1}$ by w_j^{r+1} and the bound is only larger.

To see why (4.2) is true observe that after iteration q, each non-zero entry in column j is equal to δ, and thus $w_j^{q+1} \leq d_j\delta$. Furthermore since $\delta \leq 1$, if $a_{ij}^{q+1} \neq 0$ then every nonzero entry in row i is the same, δ. Thus if a_{ij} is reduced again it is reduced to 0. The column sum must be reduced by some multiple of δ. Upon factoring out the δ from top and bottom of " $(w_j^r - w_j^{r+1})/w_j^r$ ", we see that the sum

$$\sum_{r=q+1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r}$$

may be analyzed as before in the case of integral data; $w_j^{q+1}/\delta \leq d_j$, thus

$$\sum_{r=q+1}^{v_j} \frac{w_j^r - w_j^{r+1}}{w_j^r} \leq H(d_j).$$

To obtain the bound (4.1) we solve the optimisation problem

$$z_q = \text{maximum} \sum_{r=1}^{q} \frac{w_j^r - w_j^{r+1}}{w_j^r}$$
$$\text{s/t } w_j^1 \geq \cdots \geq w_j^q \geq 1 \quad w_j^{q+1} \geq 0$$

where w_j^1 is viewed as fixed, the rest as variable. This is solved in Lemma 4.2 and z_q is shown to be converging up to $\log(w_j^1) + 1$ as $q \to \infty$ in Lemma 4.3.

Lemma 4.2.

$$z_q = q - (q-1)(w_j^1)^{-1/(q-1)}.$$

Proof. For ease of notation define $B = w_j^1$ and make the substitution $y_{q+1-r} = w_j^r$ for $r = 1, \ldots, q+1$.

$$z_q = \text{maximum} \sum_{r=1}^{q} \frac{y_r - y_{r-1}}{y_r}$$
$$\text{s/t } B \geq y_q \geq \cdots \geq y_1 \geq 1, \quad y_0 \geq 0.$$

This is equivalent to

$$z_q = q - \text{minimum} \sum_{r=1}^{q} \frac{y_{r-1}}{y_r}$$
$$\text{s/t } B \geq y_q \geq \cdots \geq y_1 \geq 1, \quad y_0 \geq 0.$$

Observe that any optimal solution will have $y_0 = 0$, $y_q = B$. We can now solve the minimization problem via a dynamic programmming recursion. Define

$$V(B,q) = \text{minimum} \sum_{r=1}^{q} \frac{y_{r-1}}{y_r}$$
$$\text{s/t } B \geq y_q \geq \cdots \geq y_1 \geq 1, \quad y_0 \geq 0.$$
$$= \underset{B \geq y \geq 1}{\text{minimum}} \left\{ \frac{y}{B} + V(y, q-1) \right\}.$$

By the comments above $V(B,1) = 0$ and we now show by induction on q that

$$V(B,q) = (q-1)B^{-1/(q-1)}. \tag{4.3}$$

Assuming (4.3) holds for $q, \ldots, 1$ we compute

$$V(B, q+1) = \underset{B \geq y \geq 1}{\text{minimum}} \left\{ \frac{y}{B} + (q-1)y^{-1/(q-1)} \right\}.$$

Taking the first derivative with respect to y and setting the result to 0 we have

$$\frac{1}{B} - \left(\frac{q-1}{q-1}\right) y^{(-1/(q-1)-1)} = 0.$$

Solving,

$$\hat{y} = B^{(q-1)/q}.$$

If $B > 1$ then $B \geq \hat{y} \geq 1$ and upon taking the second derivative we see it is always positive for $y \geq 0$, so $\hat{y}$ is a global minimum;

$$\frac{q}{(q-1)} y^{(-1/(q-1)-2)} \geq 0 \quad \text{for } y \geq 0$$

Substituting back we have

$$\begin{aligned} V(B, q+1) &= \frac{B^{(q-1)/q}}{B} + (q-1)\left(B^{(q-1)/q}\right)^{-1/(q-1)} \\ &= qB^{-1/q}. \end{aligned}$$

Thus

$$z_q = q - (q-1)(w_j^1)^{-1/(q-1)}.$$

∎

Lemma 4.3.

$$f(q) \equiv z_{q+1} - 1 = q(1 - B^{-1/q}) \uparrow \log(B) \quad \text{as } q \to \infty.$$

Proof.

$$\begin{aligned} f'(q) &= \left(1 - B^{-1/q}\right) + q(-1/q^2)B^{-1/q}\log(B) \\ &= 1 - B^{-1/q}\left(1 + \frac{\log(B)}{q}\right) > 0 \\ \text{iff} \quad B^{1/q} &> \left(1 + \frac{\log(B)}{q}\right) \\ \text{iff} \quad \frac{1}{q}\log(B) &> \log\left(1 + \frac{1}{q}\log(B)\right) \end{aligned}$$

iff $\qquad \alpha \qquad > \log(1+\alpha)$

with $\alpha = \frac{1}{q}\log(B)$ so $f' > 0$ always.

$$\lim_{q\to\infty} f(q) = \lim_{q\to\infty} \frac{1-B^{-1/q}}{1/q}$$

by l'Hopital's rule

$$= \lim_{q\to\infty} \frac{-(1/q^2)B^{-1/q}\log(B)}{-1/q^2}$$
$$= \log(B).$$

Since $f'(q) > 0$ for all q and the limit is $\log(B)$ we have then that z_q converges monotonically up to $1 + \log(B)$, as $q \to \infty$. ∎

5. Some Negative Results

A much more satisfactory result for an approximation algorithm would be one that gave a fixed bound, r, on the error independent of the size of the problem data. If no such algorithm existed one might hope to at least show that the existence of such an algorithm implied $\mathcal{P} = \mathcal{NP}$ (see [2]). Alternatively a nice extension of the above results would be to the case where negative numbers were allowed in the matrix. The following result suggests that such an extension is unlikely. This technique is analogous to a result of Garey and Johnson[2] on the maximum independent set problem.

Consider the integer programming problem

$$\begin{aligned} z_1 = \text{minimum } & cx \\ & Ax \geq b \\ & x_j = 0,1 \quad j = 1,\ldots,n \end{aligned} \tag{P$_1$}$$

where $a_{ij} \in \{0,1,-1\}, b_i \in \{0,1\}, c_j \in \{0,1\}$ for all i,j.

Theorem 5.1. Let x^* be the optimal solution to (P_1) and let $\bar{x}$ be a solution given by an approximation algorithm. Either there is a polynomial approximation scheme to solve (P_1) such that

$$\frac{c\bar{x}}{cx^*} \leq 1 + \epsilon \quad \text{for arbitrary } \epsilon > 0$$

or there does not exist any polynomial approximation scheme that gives

$$\frac{c\bar{x}}{cx^*} \leq r \quad \text{for any } r \geq 1.$$

First we need

Lemma 5.2. Given any problem of the form P_1 we can create another problem, call it P_k, of the same form with optimal value z_k such that

$$z_k = (z_1)^k.$$

Furthermore, given any solution to P_k with value v_k we can extract from it a solution to problem P_1 with value v_1 such that

$$v_1 \leq (v_k)^{\frac{1}{k}}.$$

Proof. For simplicity we start with the case $k = 2$. Consider

$$\begin{aligned} z_2 = \text{minimum} \quad & \sum_{i=1}^{n} (c_i c) x^i \\ \text{s/t} \quad & Ax^i - by_i \geq 0 \\ & Ay \geq b \\ & x^i_j = 0,1 \quad y_i = 0,1 \end{aligned} \tag{P_2}$$

where $c_i c$ is the vector c times the scalar c_i, x^i is an n-vector for $i = 1, \ldots, n$, y is an n-vector, by_i is the vector b times the scalar y_i.

It is easy to check that P_2 is a problem of the form P_1. Next, observe that if $y_i = 1$ in some solution then $Ax^i \geq by_i = b$, thus $cx^i \geq z_1$. Let $\left(\{x^i\}_{i=1}^n, y\right)$ be a feasible solution to P_2, then

$$\begin{aligned} \sum_{1 \leq i \leq n} (c_i c) x^i \geq \sum_{\substack{1 \leq i \leq n \\ y_i = 1}} (c_i c) x^i &= \sum_{1 \leq i \leq n} c_i (cx^i) y_i \\ &\geq z_1 \sum_{1 \leq i \leq n} c_i y_i \geq z_1 z_1 = (z_1)^2. \end{aligned}$$

Let y^* be the optimal solution to P_1, then we claim that $\left(\{x^i\}_{i=1}^n, y\right)$ given by

$$\begin{aligned} y &= y^* \\ x^i &= \begin{cases} y^*, & \text{if } y_i = 1, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

is an optimal solution to P_2 since,

$$\begin{aligned} \sum_{1 \leq i \leq n} (c_i c) x^i = \sum_{1 \leq i \leq n} c_i (cx^i) y_i \\ = z_1 \sum_{1 \leq i \leq n} c_i y_i \\ = (z_1)^2 \end{aligned}$$

thus $z_2 = (z_1)^2$. This proves the first part. Again let $\left(\{x^i\}_{i=1}^n, y\right)$ be any feasible solution to P_2 and let $\bar{x}$ be the vector that minimizes

$$\min\left\{cy, \min_{1\le i\le n} cx^i\right\}$$

so that $c\bar{x} \le cx^i$ and $c\bar{x} \le cy$.

$$\begin{aligned} v_2 = \sum_{1\le i\le n} c_i(cx^i) &\ge \sum_{1\le i\le n} c_i(cx^i)y_i \\ \ge \sum_{1\le i\le n} c_i(c\bar{x})y_i &\ge (c\bar{x}) \sum_{1\le i\le n} c_i y_i \ge (c\bar{x})^2 \end{aligned}$$

which implies that

$$c\bar{x} \le (v_2)^{\frac{1}{2}}.$$

This proves the lemma for $k = 2$.

We now go by induction. Assume we have the problem

$$\begin{aligned} z_k = \text{minimum } & dx \\ & Dx \ge h \\ & x_j = 0, 1 \end{aligned} \qquad (P_k)$$

The P_{k+1} problem is

$$\begin{aligned} z_{k+1} = \text{minimum } & \sum_{1\le i\le n} (c_i d)x^i \\ & Dx^i - hy_i \ge 0 \\ & Ay \ge b \\ & x_j^i = 0, 1 \quad y_i = 0, 1 \end{aligned} \qquad (P_{k+1})$$

Problem P_{k+1} is of the proper form if P_1 and P_k are. Let $\left(\{x^i\}_{i=1}^n, y\right)$ be a solution to P_{k+1}.

$$\begin{aligned} \sum_{1\le i\le n} (c_i d)x^i \ge \sum_{1\le i\le n} c_i(dx^i)y_i &\ge \sum_{1\le i\le n} c_i(z_k)y_i \\ \ge z_k \sum_{1\le i\le n} c_i y_i &\ge z_k z_1 = (z_1)^k z_1 = (z_1)^{k+1}. \end{aligned}$$

Let y^* be the optimal solution to P_1, and we claim as before that $\left(\{x^i\}_{i=1}^n, y\right)$ given by

$$y = y^*$$
$$x^i = \begin{cases} \text{"optimal solution to } P_k\text{"}, & \text{if } y_i = 1, \\ 0, & \text{otherwise,} \end{cases}$$

is optimal to P_{k+1}.

$$\begin{aligned} z_{k+1} &= \sum_{1\le i\le n} c_i(dx^i) = \sum_{1\le i\le n} c_i(dx^i)y_i = \sum_{1\le i\le n} c_i(z_k)y_i \\ &= z_k \sum_{1\le i\le n} c_i y_i = z_k z_1 = (z_1)^k z_1 = (z_1)^{k+1}. \end{aligned}$$

Now let $\left(\{x^i\}_{i=1}^n, y\right)$ be any feasible solution with value v_{k+1}. For each i let u^i be the best solution obtained to the P_1 problem from the x^i vector as before. Let u be the best among the u^i's and y so that by induction we have

$$cu^i \le \left(dx^i\right)^{1/k}$$

and

$$cu \le cu^i \qquad cu \le cy.$$

Finally,

$$\begin{aligned} v_{k+1} &= \sum_{1\le i\le n} c_i(dx^i) \ge \sum_{1\le i\le n} c_i(dx^i)y_i \ge \sum_{1\le i\le n} c_i\left(cu^i\right)^k y_i \\ &\ge (cu)^k \sum_{1\le i\le n} c_i y_i \ge (cu)^k cu = (cu)^{k+1} \end{aligned}$$

thus

$$cu \le (v_{k+1})^{1/(k+1)}.$$

∎

We can now prove the theorem. Assume we have a problem P_1 and an algorithm for which $\dfrac{c\hat{x}}{cx^*} \le r$. Let $\epsilon > 0$ fixed. Pick k large enough so that $r^{1/k} \le 1 + \epsilon$. Note that k is fixed since it depends only on the fixed ϵ. Now construct problem P_k from P_1 as above. Apply the algorithm to P_k and obtain a solution (with value v_k). From this we construct a solution to P_1 (with value

v_1). The lemma above says that $(z_1)^k = z_k$ and $(v_1)^k \leq v_k$ thus we have

$$\left(\frac{v_1}{z_1}\right)^k = \frac{(v_1)^k}{(z_1)^k} \leq \frac{v_k}{z_k} \leq r$$

which implies

$$\frac{v_1}{z_1} \leq r^{1/k} \leq 1 + \epsilon.$$

If the original problem P_1 had n variables and m constrants then the problem P_k would have $n\sum_{i=0}^{k} n^i$ variables and $m\sum_{i=0}^{k} n^i$ constraints. If our algorithm ran in $O(p(m,n))$ time and gave at worst an error of r, applying it to P_k above would give a solution procedure that ran in $O(p(mn^k, n^{k+1}))$ time and gave an error of $r^{1/k}$. Thus either there is no polynomial approximation scheme that has a fixed error bound, or there is a polynomial approximation scheme that can be adjusted to give an error bound of $1+\epsilon$ for arbitrary $\epsilon > 0$. ∎

We turn now to an even simpler problem, the set covering problem with unit costs. Presumably this is a more tractable problem. From Theorem 2.1 we know the error is bounded by $H(k)$ where the maximum number of ones in any column is k. An example where the error is exactly $H(k)$ from Johnson[3] is given below for the case $k = 3$. Here $c = (1,\ldots,1)$ and $b = (1,\ldots,1)$.

$$A = \begin{pmatrix}
1 &&&&&& 1 &&&&&&&&&& \\
1 &&&&&&&& 1 &&&&&&&& \\
1 &&&&&&&&&&& 1 &&&&& \\
& 1 &&&&& 1 &&&&&&&&&& \\
& 1 &&&&&&& 1 &&&&&&&& \\
& 1 &&&&&&&&&&& 1 &&&& \\
&& 1 &&&& 1 &&&&&&&&&& \\
&& 1 &&&&&&& 1 &&&&&&& \\
&& 1 &&&&&&&&&&& 1 &&& \\
&&& 1 &&&& 1 &&&&&&&&& \\
&&& 1 &&&&&& 1 &&&&&&& \\
&&& 1 &&&&&&&&&&& 1 && \\
&&&& 1 &&& 1 &&&&&&&&& \\
&&&& 1 &&&&&& 1 &&&&&& \\
&&&& 1 &&&&&&&&&&& 1 & \\
&&&&& 1 && 1 &&&&&&&&& \\
&&&&& 1 &&&&& 1 &&&&&& \\
&&&&& 1 &&&&&&&&&&& 1
\end{pmatrix}$$

$$x = (0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1)$$

$$x^* = (1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0)$$

It might appear that all that is needed to reduce the error substantially is to find a proper tie-breaking rule. That is, some way to decide among the columns that have the longest lengths. Unfortunately, we have

Theorem 5.3. Applying the greedy heuristic to the set covering problem with unit costs, no tie-breaking rule, i.e. a rule that chooses between columns of the same length, can guarantee an error less than $H(k) - \frac{1}{2}$.

Proof. We construct an example where the column choice is forced for the greedy algorithm and any tie-breaking rule would not change the column choices. The matrix A is constructed for the case of $k = 4$ (see next page); the extension to arbitrary k should be clear. First construct a matrix D with k columns and $k(k-1)$ rows where each column has $(k-1)$ 1's.

$$D = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

Create a block diagonal matrix with the matrix D on the diagonal. The columns of this will make up the optimal solution. The greedy heuristic covers the first k rows of each copy of D with columns of length k, the next k rows with columns of length $k-1$, ... the next to last k rows with columns of length 3 and the last k rows with columns of length 1. The entire case $k = 4$ is below. At each iteration the greedy algorithm picks a column of maximum length and no tie-breaking rule would ever be used to any advantage. ∎

$$A=\left(\begin{array}{*{31}{c}}
1&&&&&&&&&&&&1\\
&1&&&&&&&&&&&1\\
&&1&&&&&&&&&&1\\
&&&1&&&&&&&&&1\\
1&&&&&&&&&&&&&&&1\\
&1&&&&&&&&&&&&&&&1\\
&&1&&&&&&&&&&&&&&&1\\
&&&1&&&&&&&&&&&&&&&1\\
1&&&&&&&&&&&&&&&&&&&1\\
&1&&&&&&&&&&&&&&&&&&&1\\
&&1&&&&&&&&&&&&&&&&&&&1\\
&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&1&&&&&&&&&1\\
&&&&&1&&&&&&&&1\\
&&&&&&1&&&&&&&1\\
&&&&&&&1&&&&&&1\\
&&&&1&&&&&&&&&&&1\\
&&&&&1&&&&&&&&&&&1\\
&&&&&&1&&&&&&&&&&&1\\
&&&&&&&1&&&&&&&&&&&1\\
&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&&&1&&&&&&1\\
&&&&&&&&&1&&&&&1\\
&&&&&&&&&&1&&&&1\\
&&&&&&&&&&&1&&&1\\
&&&&&&&&1&&&&&&&1\\
&&&&&&&&&1&&&&&&&1\\
&&&&&&&&&&1&&&&&&&1\\
&&&&&&&&&&&1&&&&&&&1\\
&&&&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&&&&&1&&&&&&&&&&&&&&&&&&&1\\
&&&&&&&&&&&1&&&&&&&&&&&&&&&&&&&1
\end{array}\right)$$

$$x^* = (1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$$
$$x = (0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$$

References

[1] Chvátal, V. (1979). A Greedy Heuristic for the Set Covering Problem. *Math. Oper. Res.*, **4**, pp. 233–235.

[2] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freedman and Co., San Fransisco.

[3] Johnson, D. S. (1974). Approximation Algorithms for Combinatorial Problems, *J. Comput. System Sci.*, **9**, pp. 256–278.

[4] Lovász, L. (1975). On the Ratio of Optimal Integral and Fractional Covers, *Discrete Math.*, **13**, pp. 383–390.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) SOL-80-25	2. GOVT ACCESSION NO. AD-A093	3. RECIPIENT'S CATALOG NUMBER 618
4. TITLE (and Subtitle) (6) WORST CASE ANALYSIS OF GREEDY HEURISTICS FOR INTEGER PROGRAMMING WITH NON-NEGATIVE DATA.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Greg DOBSON	(15)	8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267 DE-AC03-76SF00326
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research - Dept. of the Navy 800 N. Quncy Street Arlington, VA 22217	(11)	12. REPORT DATE October 1980
		13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

WORST CASE ERROR ANALYSIS
GREEDY HEURISTIC
INTEGER PROGRAMMING
SET COVERING

$\geq \alpha =$

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We give a worst case analysis for two greedy heuristics for the integer programming problem minimize cx, $Ax \geq b$, $0 \leq x \leq u$, x integer, where the entries in A, b, and c are all non-negative. The first heuristic is for the case where the entries in A and b are integral, the second only assumes the rows are scaled so that the smallest nonzero entry is at least 1. In both cases we compare the ratio of the value of the greedy solution to that of the integer optimal. The error bound grows logarithmically in the maximum column sum of A for both heuristics.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

408765

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)